

Задача А. Интернет-магазин

При покупке товаров в интернет-магазинах все выбранные товары складываются в корзину. При этом покупатели могут забыть добавить какой-нибудь из нужных им товаров.

Для того, чтобы покупатели остались довольны покупкой, а магазин получил больше прибыли существует механизм рекомендаций, который определяет, какие товары обычно покупают вместе с набором уже выбранных. Например, если покупатель положил в корзину ластик, то, наверняка, ему также понадобится карандаш.

Вам необходимо разработать сервис рекомендаций, который по истории предыдущих заказов разработает для покупателя рекомендации, основанные на текущем состоянии его заказа.

Рекомендации должны быть двух типов: «с этим товаром всегда берут следующие товары» и «с этим товаром часто берут следующие товары». При этом «часто» понимается как 50% и более. Например, если покупатель хочет купить два товара А и В, а предыдущие заказы были вида (А, D), (В, С, Е), (С, F), (С, Е, F, G) и (А, В, С, Е), то товары С и Е надо рекомендовать как те, что покупается всегда (вместе с товаром В), а D — как тот, что покупается часто (50% случаев заказов с товаром А).

Если товар всегда покупался с одним из заказанных, то необходимо включить в число часто покупаемых и те, которые часто встречаются с этим товаром (не менее чем в 50% случаев) в ранее сделанных заказах. Таким образом, дополнительно к товарам покупаемым часто, добавится товар F, который часто покупается с товаром С. Товар G рекомендовать не нужно, т.к. он встречается меньше, чем в 50% заказов вместе с товаром С.

Не нужно рекомендовать товары, которые уже выбрал покупатель. Если товар можно рекомендовать как «часто» и «всегда», то следует рекомендовать его только как «всегда».

Входные данные

В первой строке входного файла записывается количество старых заказов N . Следующие N строк содержат названия товаров, сделанных в этом заказе. Каждое название состоит из одного слова, слова разделены пробелами.

В следующей строке содержится описание текущего заказа: набор названий товаров, разделенных пробелами.

Результат

Для текущего заказа выведите две строки. Первая строка должна содержать названия товаров, которые всегда покупаются вместе с товарами из текущего заказа, вторая — с названиями часто покупаемых товаров (соответственно определению часто покупаемых товаров из условия задачи). Слова следует разделять пробелами. Порядок вывода не важен.

Пример

Входные данные	Результат
5 A D B C E C F C E F G A B C E A B	C E D F

Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся только если сформированные вами рекомендации удовлетворяют всем требованиям, описанным в условии, для каждого из блоков. Проверка осуществляется в режиме on-line (результат виден сразу).

Максимальная оценка за второй тест — 70 баллов. Каждый неверно рекомендованный товар (лишний или недостающий) уменьшает оценку на 10 баллов. Оценка не может стать меньше нуля. Проверка осуществляется в режиме off-line (после окончания тура).

Задача В. Схема

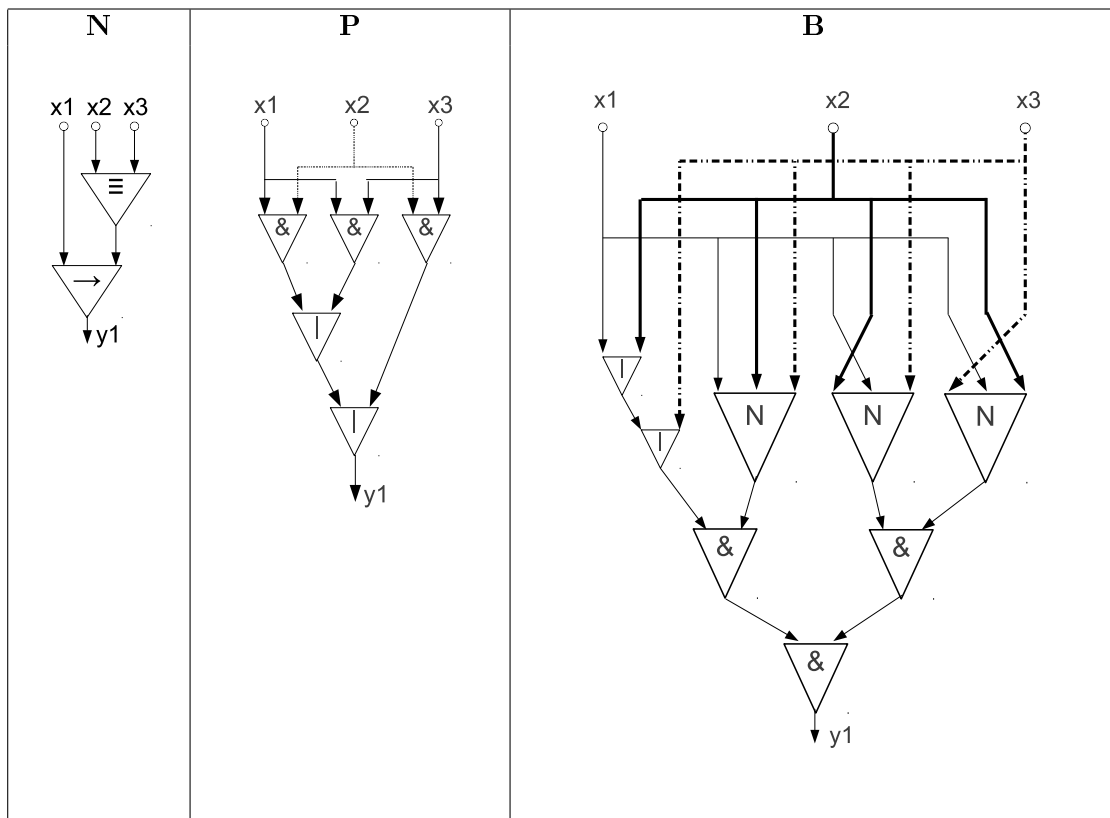
Схемы из функциональных элементов предназначены для реализации логических формул. Для схемы, как и для формулы, можно составить таблицу истинности. Приведем значения бинарных функций для всех возможных пар параметров:

Обозначение	Название	00	01	10	11
&	И	0	0	0	1
	Или	0	1	1	1
≡	Тождественно равно	1	0	0	1
→	Импликация	1	1	0	1

Значком \sim обозначается отрицание. Приведем пример схемы и таблицы истинности для нее:

Схема	Таблица истинности	Ответ																																												
	<table border="1"> <thead> <tr> <th>x1</th> <th>x2</th> <th>x3</th> <th>y1, y2</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1 0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1 0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0 1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0 0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0 1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0 0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0 1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0 0</td></tr> </tbody> </table>	x1	x2	x3	y1, y2	0	0	0	1 0	0	0	1	1 0	0	1	0	0 1	0	1	1	0 0	1	0	0	0 1	1	0	1	0 0	1	1	0	0 1	1	1	1	0 0	<table border="1"> <tbody> <tr><td>1 0</td></tr> <tr><td>1 0</td></tr> <tr><td>0 1</td></tr> <tr><td>0 0</td></tr> <tr><td>0 1</td></tr> <tr><td>0 0</td></tr> <tr><td>0 1</td></tr> <tr><td>0 0</td></tr> </tbody> </table>	1 0	1 0	0 1	0 0	0 1	0 0	0 1	0 0
x1	x2	x3	y1, y2																																											
0	0	0	1 0																																											
0	0	1	1 0																																											
0	1	0	0 1																																											
0	1	1	0 0																																											
1	0	0	0 1																																											
1	0	1	0 0																																											
1	1	0	0 1																																											
1	1	1	0 0																																											
1 0																																														
1 0																																														
0 1																																														
0 0																																														
0 1																																														
0 0																																														
0 1																																														
0 0																																														

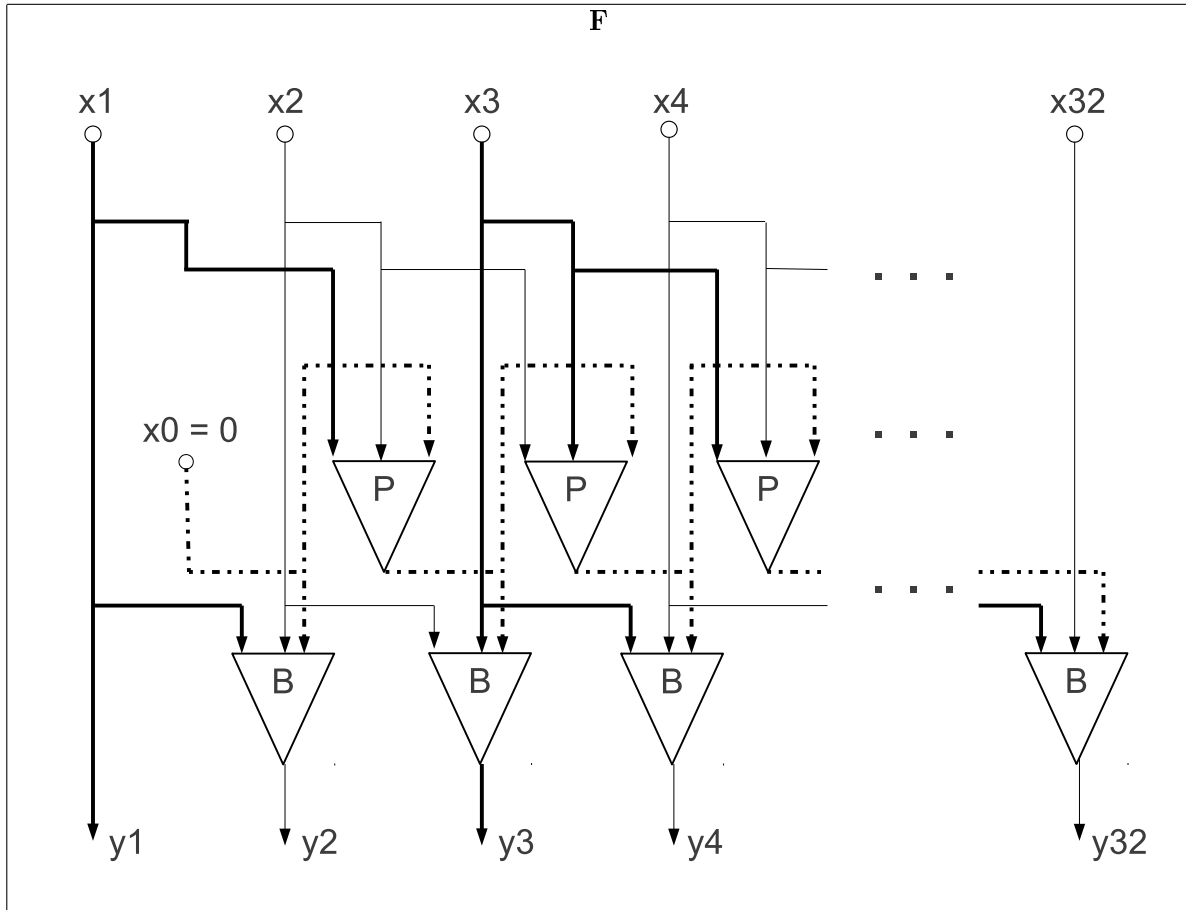
Ответом для схемы являются столбцы результата (y_1, y_2, \dots) ее таблицы истинности для всевозможных значений x_1, x_2, \dots , перебирающихся в том же порядке, как в примере. Для упрощения чтения и реализации схемы возможно объединение нескольких логических элементов в блок, каждый блок мы будем обозначать большой латинской буквой.



В первом тесте вам предстоит рассчитать сначала блок, обозначаемый «В». В качестве ответа необходимо сдать 8 чисел 1 или 0, разделенных пробелами — результат, выдаваемый схемой для

всех возможных наборов входных данных. x_1 , x_2 и x_3 должны перебираться в том же порядке, как и в примере.

Во втором тесте вам необходимо рассчитать значения для 7 заданных во входном файле наборов параметров для схемы «F». В качестве ответа необходимо сдать 7 наборов из 32 чисел 1 и 0 — результат работы схемы для заданных наборов входных значений. Числа в одном наборе должны разделяться пробелами, наборы должны разделяться переводами строки.



Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся только если все значения определены правильно. Проверка осуществляется в режиме on-line (результат виден сразу).

Максимальная оценка за второй тест — 70 баллов. Каждый неверно рассчитанный набор уменьшает оценку на 10 баллов. Оценка не может стать меньше нуля. Проверка осуществляется в режиме off-line (после окончания тура).

Задача С. LZ78

В 1978 году Абрахам Лемпел и Якоб Зив разработали алгоритм сжатия LZ78, модификации которого используются и сейчас для zip, gif и других популярных форматов общих и специализированных архивов.

Рассмотрим, как происходит сжатие строки, состоящей из больших латинских букв. В начале работы словарь состоит из пустого слова с номером 0, а также из всех больших латинских букв, расположенных по алфавиту (эти однобуквенные слова имеют номера от 1 до 26).

Алгоритм считывает символы до тех пор, пока накапливаемая подстрока входит целиком в одно из слов словаря. Как только эта строка перестанет соответствовать хотя бы одному слову из словаря, алгоритм генерирует код, состоящий из номера строки в словаре, которая до последнего введенного символа содержала входную строку, и символа, нарушившего совпадение. Затем в словарь добавляется введенная подстрока (с номером, на единицу большим последнего слова в словаре). Если в конце алгоритма мы не находим символ, нарушивший совпадения, то тогда мы выдаем код в виде (номер строки в словаре без последнего символа, последний символ).

Рассмотрим, как будет выглядеть словарь для фразы «АВАВВАВВВАВААВ».

Номер предыдущего слова	Добавленная буква	Добавленное слово	Номер добавленного слова
1	В	АВ	27
27	В	АВВ	28
28	В	АВВВ	29
27	А	АВА	30
1	В	АВ	31

Сжатие достигается за счет того, что для восстановления исходной строки достаточно знать только два первых столбца этой таблицы. Вам предстоит распаковать данные, сжатые с помощью алгоритма LZ78.

Входные данные

Во входном файле записаны строки, каждая из которых содержит число — номер предыдущего слова и добавляемую к нему букву. Гарантируются, что данные корректны, число и буква разделены символом табуляции.

Результат

В качестве ответа необходимо сдать распакованную строку.

Пример

Входные данные	Результат
1 В 27 В 28 В 27 А 1 В	АВАВВАВВВАВААВ

Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся только если строка полностью совпадает с правильным ответом. Проверка осуществляется в режиме on-line (результат виден сразу).

Второй тест оценивается в 70 баллов. Эти баллы ставятся только если строка полностью совпадает с правильным ответом. Проверка осуществляется в режиме off-line (после окончания тура).

Задача D. Сложившиеся проблемы

Недавно на сервере с архивами олимпиад произошёл сбой. В результате оказалась утеряна существенная часть накопившихся тестов и решений задач. К счастью, часть данных всё-таки удалось спасти. Так, по одной из задач, хранившейся под названием 'Неправильное A+B', жюри удалось восстановить:

- Часть тестов, в которых некоторые символы оказались заменены на знак '?'
- Некоторые решения участников, присланные на сервер
- Файлы с ответами, полученными при запуске восстановленных решений участников (все восстановленные решения давали одинаковый ответ)

Также известно, что каждый из тестов в задаче состоял ровно из двух целых неотрицательных чисел. Числа во входных файлах были записаны в различных строках, состояли из не более чем 2000 цифр каждое и могли иметь ведущие нули.

Ниже приведены тексты восстановленных решений участников на различных языках программирования:

C++ (GNU C++)	Pascal (FreePascal)
<pre>#include <iostream> #include <string> using namespace std; int main() { string a, b; char k = 0; cin >> a >> b; if (a.size() < b.size()) { swap(a, b); } for (int i = 0; i < b.size(); ++i) { a[i] = (a[i] - '0') + (b[i] - '0'); } for (int i = b.size() - 1; i >= 0; --i){ a[i] += k; k = a[i] / 10; a[i] = a[i] % 10 + '0'; } if (k > 0) { a = "1" + a; } cout << a << endl; return 0; }</pre>	<pre>var a,b,c: string; i, k: integer; begin readln(a); readln(b); if length(b) > length(a) then begin c := a; a := b; b := c end; for i := 1 to length(b) do a[i] := chr(ord(a[i])+ord(b[i])-ord('0')); k := 0; for i := length(b) downto 1 do begin a[i] := chr(ord(a[i])+k); k := (ord(a[i])-ord('0')) div 10; a[i] := chr((ord(a[i])-ord('0')) mod 10+ord('0')); end; if k > 0 then a := '1' + a; writeln(a) end.</pre>

Python3:

```
a = list(map(int, input()))
b = list(map(int, input()))
k = 0;
if len(a) < len(b):
    a, b = b, a
for i in range(len(b)):
    a[i] = a[i] + b[i]
for i in range(len(b) - 1, -1, -1):
```

```
a[i] += k
k = a[i] // 10
a[i] = a[i] % 10
if k:
    a = [1] + a
a = ''.join(map(str, a))
print(a)
```

К сожалению, этих данных оказалось недостаточно для того, чтобы однозначно восстановить архив с тестами. Несмотря на это, жюри просит Вас найти хоть какой-нибудь способ заменить возникшие в тестах символы '?' на десятичные цифры таким образом, чтобы результат работы восстановленных решений участников совпадал с содержимым восстановленных файлов с ответами.

Входные данные

Первая строка входного файла содержит количество тестов, которые Вам необходимо восстановить. Далее следуют блоки с описаниями отдельных тестов, составленные из трёх строк каждый и разделённые дополнительными пустыми строками.

В каждом из блоков первые две строки содержат спасённое содержимое тестов, а третья строка — результат работы решений участников на изначальных входных данных.

Результат

Для каждого блока с описанием теста в отправляемый файл поместите два числа, полученных Вами в процессе замены символов '?' на десятичные цифры.

Ответы на различные блоки следует разделять пустой строкой.

Примеры

Входные данные	Результат
2	10
??	9
?	0392
100	0955
?39?	
??55	
1347	

Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся только если сформированные Вами тесты удовлетворяют всем требованиям, описанным в условии, для каждого из блоков. Проверка осуществляется в режиме on-line (результат виден сразу).

Максимальная оценка за второй тест — 70 баллов. Каждый неверно восстановленный блок уменьшает оценку на 7 баллов. Оценка не может стать меньше нуля. Проверка осуществляется в режиме off-line (после окончания тура).

Задача Е. Банк

Михаил заработал свой первый миллион рублей. Прямо сейчас этот миллион ему не нужен, поэтому он решил положить деньги на депозит в банк «Лесенка».

Приведем информацию о вкладах в банке:

- Банк предлагает следующие процентные ставки по вкладам: на вклад от 3 месяцев до 5 месяцев — 4% годовых, на вклад от 6 месяцев до 11 месяцев — 6.5% годовых, на вклад на 12 месяцев — 9.5% годовых.
- Проценты начисляются ежемесячно, например, по окончании первого месяца вклада, открытого под 4% годовых, начисляется $4\%/12 = (1/3)\%$ суммы вклада. Эта сумма прибавляется к сумме вклада, и в следующем месяце проценты начисляются также и на нее. Таким образом, вклад на год приносит чуть больше 9.92% от суммы вклада. Это называется «ежемесячная капитализация процентов».
- Досрочное закрытие вклада невозможно.
- При открытии вклада клиенту начисляется бонус, равный 1.5% от суммы вклада. Бонус сразу же приписывается к сумме вклада.
- Минимальная сумма вклада равна 30 000 рублей (без учета 1.5% бонуса).
- Возможно пополнение вклада. При пополнении вклада за 3 или более месяцев до его окончания сразу же начисляется бонус 1.5% от суммы пополнения и приписывается к сумме вклада.
- В банке можно открыть не более 6 вкладов.
- По окончании срока вклада вся сумма обязательно должна быть переведена в наличные. При этом происходит округление до копеек по математическим правилам.

Для упрощения можно считать, что все месяцы в году одинаковой длины и равны $1/12$ года.

Помогите Михаилу заработать как можно больше денег. Необходимо разработать два плана: на 1 год (12 месяцев) и на 4 года (48 месяцев). В начале имеется 1 000 000 рублей наличными. Все действия совершаются после прохождения целого числа месяцев от начала.

Результат

Для описания плана инвестирования предусмотрено две команды:

- OPEN <момент времени> <номер вклада> <срок вклада в месяцах> <сумма вклада> — открыть вклад. Момент времени — это количество прошедших месяцев, в начальный момент времени это число равно нулю. Номер вклада может быть от 1 до 6, при этом не должно быть открытого вклада с таким номером, который еще не окончился. Срок вклада задается в месяцах и может быть любым целым числом от 3 до 12. По окончании срока вклада все деньги вместе с начисленными процентами и бонусами будут переведены в наличные. Сумма должна быть указана с точностью до копеек.
- ADD <момент времени> <номер вклада> <сумма пополнения> — пополнить вклад. Момент времени — это количество прошедших месяцев. Номер вклада может быть от 1 до 6, вклад с таким номером должен быть открыт и не должен заканчиваться в текущий момент времени. Сумма должна быть указана с точностью до копеек. Пополнять вклад можно только из наличных денег.

Команды должны быть упорядочены по времени.

В первую подзадачу необходимо сдать последовательность команд, позволяющую получить наибольшую сумму за 12 месяцев, во вторую — за 48 месяцев. Все вклады должны закрыться к моментам времени 12 и 48 соответственно.

Пример

Входные данные	Результат
6 месяцев, сумма 100000	OPEN 0 1 3 30000.00 OPEN 0 2 6 70000.00 ADD 3 2 755.52 OPEN 3 1 3 30000.00

В этом примере в момент времени 0 открывается два вклада: с номером 1 на 3 месяца и суммой 30 000 рублей и с номером 2 на 6 месяцев и суммой 70 000 рублей. В момент времени 3 вклад номер 1 заканчивается, вся сумма с процентами (30755.52) автоматически переводится в наличные. Команда ADD добавляет к уже открытому вкладу номер 2 755.52 рублей, а последняя команда открывает вклад в момент времени 3, сроком на 3 месяца и суммой 30000 рублей. Это решение не является оптимальным.

Все вклады заканчиваются в момент времени 6, все деньги переводятся в наличные и сумма составляет 104925.52 (30755.52 от вклада номер 1 и 74170.00 от вклада номер 2).

Приведем таблицу изменения сумм на вкладах (для удобства чтения промежуточные суммы в таблице округлены до копеек

):

Момент	Вклад	Сумма	Пополнение	Проценты	Комментарий
0	1	30450	30450	0	
0	2	71050	71050	0	
1	1	30551.50	0	101.50	
1	2	71434.85	0	384.85	
2	1	30653.34	0	101.84	
2	2	71821.79	0	386.94	
3	1	30755.52	0	102.18	Вклад закрывается
3	1	30450	30450	0	Открывается новый
3	2	72997.68	766.85	389.03	Пополнение 755.52 + бонус
4	1	30551.50	0	101.50	
4	2	73372.98	0	395.30	
5	1	30653.34	0	101.84	
5	2	73770.41	0	397.44	
6	1	30755.52	0	102.18	Вклад закрывается
6	2	74170.00	0	399.59	Вклад закрывается

Данная задача оценивается следующим образом:

Максимальная оценка за первый тест — 30 баллов. Балл рассчитывается по формуле $30 \times \frac{partAns - 1115736}{juryAns - 1115736}$, где *juryAns* — максимальная возможная сумма, а *partAns* — сумма, полученная Вашим решением. Оценка не может стать меньше нуля. Проверка осуществляется в режиме on-line (результат виден сразу).

Максимальная оценка за второй тест — 70 баллов. Балл рассчитывается по формуле $70 \times \frac{partAns - 1549694}{juryAns - 1549694}$, где *juryAns* — максимальная возможная сумма, а *partAns* — сумма, полученная Вашим решением. Оценка не может стать меньше нуля. Проверка осуществляется в режиме off-line (после окончания тура).

Задача F. Азбука

У Вани есть магнитная азбука, состоящая из букв английского алфавита. Каждая буква в ней присутствует только один раз.

Помогите Ване составить самую длинную фразу из тех слов, которые он считает английскими. Английский язык Ваня знает плохо, поэтому его устроит любой набор из его списка слов. Длина фразы определяется количеством букв, используемых при ее составлении.

Входные данные

В первой строке входного файла записано число N — количество слов, которые Ваня считает английскими. В каждой из следующих N строк записано одно слово.

Результат

В первую строку выходного файла выведите количество букв, использованных при построении фразы. Во вторую строку — саму фразу, разделяя слова в ней пробелами.

Пример

Входные данные	Результат
9 and byn extra for just liz owen packed quit	19 just packed byn for liz

Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся, только если предложенную вами фразу можно собрать из азбуки, она состоит из Ваниных слов, а ее длина максимально возможная. Проверка осуществляется в режиме on-line (результат виден сразу).

Максимальная оценка за второй тест — 70 баллов. Баллы ставятся только если предложенную вами фразу можно собрать из азбуки, и она состоит из Ваниных слов. Фраза максимальной длины оценивается в 70 баллов. Если букв во фразе меньше, то оценка уменьшается на 15 баллов за каждую недостающую букву. Оценка не может стать меньше нуля. Проверка осуществляется в режиме off-line (после окончания тура).

Задача G. Нормальные алгоритмы Маркова

Нормальный алгоритм Маркова (или, как именовал сам автор, нормальный алгоритм) — один из стандартных способов задавать алгоритмы, записывая их в крайне простой текстовой форме.

Марковский алгоритм принимает на вход текст и изменяет его определённым образом. Описание марковского алгоритма состоит из нескольких правил *подстановки*, последовательно применяемых к тексту. Каждое правило выглядит как "образец \rightarrow замена", где "образец" и "замена" — некоторые строки. Правая часть правила может быть пустой строкой.

Исполнение марковского алгоритма происходит по шагам. На каждом шаге мы выбираем **первое** правило подстановки, левая часть которого фигурирует в тексте, находим её самое раннее вхождение в текст, и заменяем это вхождение на строку из правой части правила. Процесс повторяется до тех пор, пока хотя бы одно правило подстановки применимо к тексту. Текст, получившийся в конце — результат работы марковского алгоритма.

В качестве примера рассмотрим следующий алгоритм, состоящий из трёх правил подстановки:

1. !0 \rightarrow 0!!
2. 1 \rightarrow 0!
3. 0 \rightarrow (здесь в правой части стоит пустая строка, что означает, что вхождение строки "0" просто исключается)

Если запустить этот алгоритм на тексте "101", то изменения будут происходить следующим образом: $\underline{1}01 \xrightarrow{2} 0!\underline{0}1 \xrightarrow{1} 00!!\underline{1} \xrightarrow{2} 00!!\underline{0}! \xrightarrow{1} 00!\underline{0}!!! \xrightarrow{1} 000!!!! \xrightarrow{3} 00!!!! \xrightarrow{3} 0!!!! \xrightarrow{3} !!!!!$ (над стрелочками указаны номера применяемых подстановок). Можно убедиться, что эта программа выписывает восклицательные знаки в количестве, равном двоичному числу, поданному ей на вход.

Удивительный факт заключается в том, что марковские алгоритмы *полны по Тьюрингу*, что означает, что любую программу, которую можно написать на «мощном» языке программирования, таком как Java или C++, можно реализовать и с помощью марковского алгоритма. В качестве иллюстрации этому мы предлагаем вам вычислить результат работы нескольких марковских алгоритмов на разных наборах входных данных.

Входные данные

В этой задаче тест состоит из двух файлов: программы и набора входных данных к ней. Программа состоит из подстановок в формате, описанном в условии задачи. В программах могут встречаться комментарии, начинающиеся с символа #, которые следует игнорировать.

Набор входных данных состоит из нескольких строк, которые надо подать на ввод соответствующему марковскому алгоритму.

Результат

Вы должны сдать в тестирующую систему файл, содержащий результаты работы марковского алгоритма на наборе входных данных. В вашем файле должно быть столько же строк, сколько и во входных данных, строки должны следовать в том же порядке, что и входные данные для алгоритма.

Пример

Входные данные	Результат
<pre>алгоритм ----- # алгоритм разбора двоичного числа !0 -> 0!! 1 -> 0! 0 -> # конец алгоритма вывод ----- 1 101 1100</pre>	<pre>! !!!! !!!!!!!!!!!!</pre>

Данная задача оценивается следующим образом:

Первый тест оценивается в 30 баллов. Эти баллы ставятся только если вывод полностью совпадает с правильным ответом. Проверка осуществляется в режиме on-line (результат виден сразу).

Второй тест оценивается в 70 баллов. Эти баллы ставятся только если вывод полностью совпадает с правильным ответом. Проверка осуществляется в режиме off-line (после окончания тура).